National Défense
Defence nationale

GENERATION OF DELAYED REPLICAS OF A MAXIMAL-LENGTH SEQUENCE USING HARDWARE LOGIC

by

Claude Belisle and Pierre Maltais

DEFENCE RESEARCH ESTABLISHMENT OTTAWA
TECHNICAL NOTE 88-08

Canada

August 1987
Ottawa

National Defence / Défense nationale

GENERATION OF DELAYED REPLICAS OF A MAXIMAL-LENGTH SEQUENCE USING HARDWARE LOGIC

by

Claude Belisle and Pierre Maltais
EHF Satcom Section
Electronics Division

DEFENCE RESEARCH ESTABLISHMENT OTTAWA
TECHNICAL NOTE 88-08

PCN
D6470-0459B

August 1987
Ottawa

ABSTRACT

Pseudonoise (PN) sequences of maximal length (m-sequences) are widely used in a number of applications in communication systems. Very often a need arises to rapidly access any position within the sequence. Moreover, it is important to be able to predict the time required to execute the transition from one state of the sequence to any other. Presented in this paper, is a method that will give access to any one of the 2^N-1 positions of an m-sequence within N clock pulses, N being the number of stages in the shift register used to produce the sequence. The method is easily implemented on hardware logic circuitry and requires no software operation.

RESUME

Les séquences pseudo-aléatoires (PN) de longueur maximale sont utilisées à plusieurs fins dans bon nombre de systèmes de communication. Très souvent il est nécessaire d'accéder rapidement à une position quelconque à l'intérieur de la séquence. De plus, il est important de pouvoir prédire l'instant où le déplacement sera complété. Cet article présente une méthode permettant d'accéder n'importe laquelle des 2^N-1 positions d'une séquence maximale en deçà de N pulses de synchronisation, N étant le nombre de paliers contenus dans le régistre à déplacement utilisé pour produire la séquence. Cette méthode est facilement réalisable à l'aide de circuits logiques et ne requiert aucun logiciel.

TABLE OF CONTENTS

		PAGE
	ABSTRACT/RESUME	iii
	TABLE OF CONTENTS	v
1.0	INTRODUCTION	1
2.0	DESIGN CONCEPT	2
3.0	DESCRIPTION OF THE PROPOSED METHOD	7
4.0	APPLICATIONS	10
5.0	CONCLUSION	12
	APPENDIX A	13
	BIBLIOGRAPHY	20

1.0 INTRODUCTION

The generation of maximal length sequences (MLS or m-sequences) of pseudorandom numbers (PN) is an important field within communications . Their quasi-randomness and associated properties make them very attractive for a number of applications [1,2]. These m-sequences, of period 2^N-1, may be generated using an N stage shift register with suitable feedback connections and modulo-2 additions.

In addition to the basic capability of generating the m-sequences, communications systems require two other important attributes; the ability to move forward or backward within the sequence (usually expressed in clock cycles) and the ability to rapidly accomplish this permutation.

Described in this paper is a new technique to achieve the attributes mentioned above. A simple algorithm, easily implemented in hardware logic and requiring minimal computational time, is developed. Section 2 serves as a review of the design concepts involved in this problem along with a discussion on past attempts to solve it. Section 3 gives a detailed description of this new approach and presents an algorithm for the generation of delayed versions of an m-sequence. Section 4 indicates a few applications where such a method could be very useful. A partial example is then developed in Appendix A.

2.0 DESIGN CONCEPTS

An N stage shift register producing a maximal length sequence is mathematically represented by a generating polynomial of degree N of the general form:

$$P(x) = a_0 + a_1x + a_2x^2 + \ldots + a_Nx^N \qquad (1)$$

The degree N defines the number of stages in the shift register and the coefficients $a_k \xi (0,1)$ defines the feedback connections.

There are two main approaches towards the implementation of the generating polynomial in a linear feedback shift register configuration, the Fibonacci generator and the Galois generator. Figure 1 shows the general construction of these two generators. The Fibonacci configuration is the usual approach chosen for MLS production due to its simplicity of implementation. Each stage of the register is made to correspond to a different power of x in the polynomial. The feedback connections determine the bit pattern of the sequence. Extensive work has been done in order to evaluate the combinations of connections that produce maximal length sequences.

The output of the generator, which forms the pseudorandom sequence, may be taken from any stage of the shift register. If the Fibonacci generator is considered, it can be noticed that for every clock pulse, the sequence is shifted unmodified, from one stage to the next, offering a certain memory to the operation. If the reference output is chosen as being the content of the

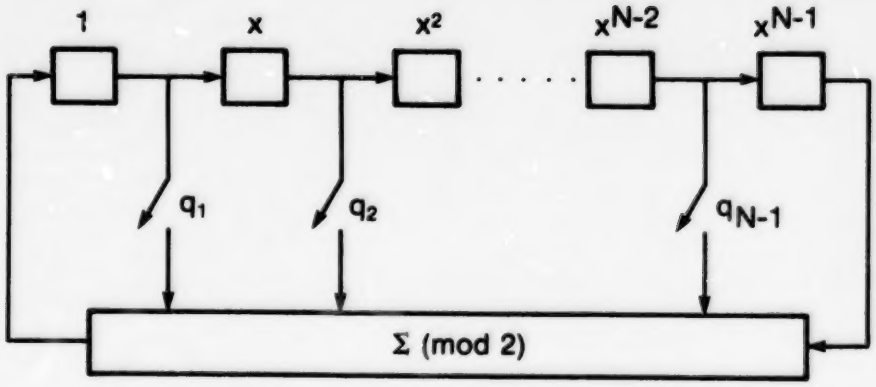


Figure 1 a) Representation of a feedback shift register in the Fibonacci configuration. The positions of the switches q's are determined from the coefficients of the generating polynomial.

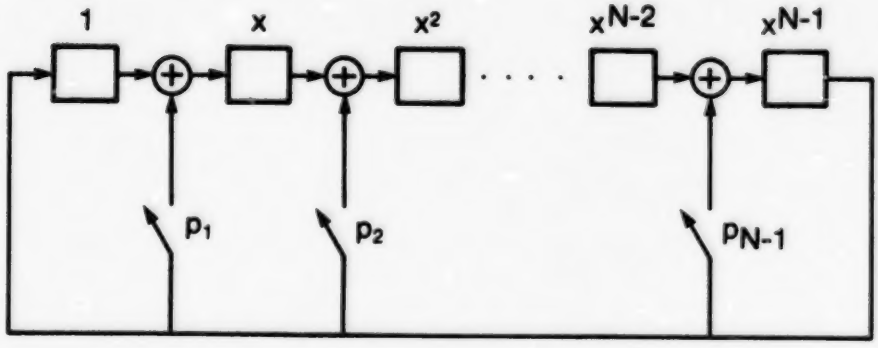


Figure 1 b) Representation of a feedback shift register in the Galois configuration. The positions of the switches p's are determined from the coefficients of the generating polynomial.

first stage, it follows that versions of the sequence delayed by 0 to N-1 pulses are directly available by tapping the proper stage of the shift register. However, there are 2^N-1-N other possible versions in a MLS which are not directly available. Suppose there is a need to delay the sequence by J bits (N-1 < J < 2^N-1), or equivalently, a need to know the state of x^J. One way to obtain this information is to simply add J+1-N stages at the end of the register until x^J is physically represented, since after J clock pulses

the initial content of the first stage of the Fibonacci generator will be present in stage J. However, this is not practical for large values of N. For example, if N equals 100, there is a possibility of over 1.26×10^{30} delays (2^N-1).

This physical difficulty may be overcome with a mathematical approach which makes use of the Galois field (GF) theory, a most useful tool in maximal length sequence analysis.

Let α be a root of the polynomial described by equation (1) such that $P(\alpha)=0$. It can be verified that the implementation of the Galois generator, as shown in figure 1, results in a multiplicative circuit by α [3]. That is, if α is initially introduced into the generator, α^2 will be present after one clock pulse, α^3 after two clock pulses and so on. Since the sequence is periodic with period of 2^N-1, it follows that $\alpha^{2^N-1} = 1$, $\alpha^{2^N} = \alpha$ ... Moreover, since the mathematical operations are part of the field $GF(2^N)$, every power of α must be representable by a combination of the basic elements of the field i.e. by a polynomial of degree less than N, N being the degree of the generating polynomial. For example, if a is a root of $P(x) = 1 + x + x^4$ in $GF(2^4)$, then:

$$P(\alpha) = 1 + \alpha + \alpha^4 = 0 \qquad (2)$$

Where the mathematical operation + corresponds to a modulo-2 addition in $GF(2^N)$. The operation * corresponds to the multiplication operation in $GF(2^N)$.

Equation 2 may then be rewritten as:

$$\alpha^4 = 1 + \alpha \tag{3}$$

Then using substitution, α^6 is found to be:

$$\begin{aligned} \alpha^6 &= \alpha^2 * \alpha^4 \\ &= \alpha^2 * (1+ \alpha) \\ &= \alpha^2 + \alpha^3 \end{aligned} \tag{4}$$

Note that the derivation of the end result reduces to a modulo-P(x) operation. Table 1 gives an exhaustive description of all 2^N-1 polynomial representation of the powers of α, as well as their corresponding binary values as produced by the Galois generator of $P(x) = 1 + x + x^4$.

Powers of α	Polynomial representation	Corresponding binary values
α^0	1	1 0 0 0
α^1	α	0 1 0 0
α^2	α^2	0 0 1 0
α^3	α^3	0 0 0 1
α^4	$1 + \alpha$	1 1 0 0
α^5	$\alpha + \alpha^2$	0 1 1 0
α^6	$\alpha^2 + \alpha^3$	0 0 1 1
α^7	$1 + \alpha + \alpha^3$	1 1 0 1
α^8	$1 + \alpha^2$	1 0 1 0
α^9	$\alpha + \alpha^3$	0 1 0 1
α^{10}	$1 + \alpha + \alpha^2$	1 1 1 0
α^{11}	$\alpha + \alpha^2 + \alpha^3$	0 1 1 1
α^{12}	$1 + \alpha + \alpha^2 + \alpha^3$	1 1 1 1
α^{13}	$1 + \alpha^2 + \alpha^3$	1 0 1 1
α^{14}	$1 + \alpha^3$	1 0 0 1

Table 1, Representation of GF(16) $P(x) = 1 + x + x^4$

The above narrative illustrates the principle of the Galois field theory that states that any delayed replicas of the m-sequence may be described by a polynomial of degree less than N. In terms of the m-sequence generator, this suggests that x^J, the delayed version of the Fibonacci generator, may be obtained by combining the output of appropriate stages

ranging from 1 to x^{N-1} i.e. with proper modulo-2 addition of selected stages of the shift register.

Several techniques have already been proposed to determine the tap connections needed to obtain a desired delayed version of an m-sequence[4-8]. However, for complex polynomials and large delays, these methods demand lenghty calculation and therefore generally requires the use of a computer. Moreover, none of the methods predict the number of operations required to obtain the desired result and none gives an estimation of the computation time. Blake and Mark [9] followed yet another approach to solve the problem. They implemented an algorithm using matrix multiplication. Although the computation time required to perform the transition became independent of the size of the delay to implement, the number of operations required to perform the matrix multiplications was still too high to be time efficient.

The method proposed in this report has the advantages of being simple, easily implemented on hardware logic circuitry (no software operations required, no computer needed) and very fast to execute. Any state of the sequence, from 0 to 2^N-1, may be attained within N clock pulses, where N is the number of stages in the shift register. The time required to reach the desired state can then be accurately predicted. For instance, an m-sequence generated with 100 registers connected in a feedback mode and clocked at 10 MHz has a period of 4×10^{15} years. Conceptually, with this method, any state within this m-sequence can be reached in 10 microseconds. The use of this method in a spread spectrum system could help to minimize the synchronization time.

3.0 DESCRIPTION OF THE PROPOSED METHOD

As previously stated, the objective is to find the state of x^J, the state of the m-sequence from the Fibonacci generator delayed by J clock pulses or equivalently, the state the sequence would be in after 2^N-1-J clock cycles. The problem lies in how to evaluate which of the N stages of the shift register should be combined in a modulo-2 adder to get x^J.

Exploiting the properties of Galois fields, it is found that the taps needed to produce a shift in the m-sequence by J clock pulses are given by the binary representation of α^J, where α is the characteristic root of the generating polynomial in the Galois field. The key point in the development is how to rapidly obtain α^J without having to produce tables similar to table 1 or to create a GF shift register and wait for J clock pulses.

Since J is bounded by (2^N-1), the delay can be represented by a binary N-tuple,

$$J = (j_0, j_1, j_2, \cdots, j_{N-1}) \tag{5}$$

where j_k equals 0 or 1. Equation 5 is mathematically expanded as follows:

$$J = j_0 + 2j_1 + 4j_2 + \ldots + 2^{N-1}j_{N-1} = \sum_{k=0}^{N-1} (2^k)j_k \tag{6}$$

The expression for α^J can then be written as:

$$\alpha \exp(J) = \alpha \exp(j_0, j_1, \ldots, j_{N-1})$$

$$\alpha^J = (\alpha)^{j_0} \cdot (\alpha^2)^{j_1} \cdot (\alpha^4)^{j_2} \cdots (\alpha^{2^{N-1}})^{j_{N-1}} \tag{7}$$

In this form, the computation of α^J and consequently the evaluation of the required taps may be performed quite simply. Equation 7 shows that α^J may be decomposed in product terms. For each exponent j_k corresponds a different power of α, (α^{2^K}), which is seen as being the square of the term corresponding to the previous exponent j_{k-1}, ($\alpha^{2^{K-1}}$). Each time a new factor ($\alpha^{2^{'}}$) is evaluated, the previous product is multiplied by either this term or by 1, according to the corresponding value $j_k \in (0,1)$, element in the binary representation of the desired shift J.

For the complete calculation, all that is needed is a squaring device and a multiplicative circuit. These two components may easily be implemented with hardware logic. The squaring device, used to sequentially evaluate the various elements α^{2^K}, is produced with an N-stage shift register. The multiplicative circuit performs a parallel multiplication and requires a 2N stage register. The two desired operations are concurrently performed within the same clock pulse. Since J is represented by an N-tuple, it follows that the evaluation of α^J requires only N clock pulses and this, independently of the value of J. The delayed version may then be obtained as quickly.

Figure 2 presents the algorithm that enables α^J to be evaluated from equation 7. The vectors $\underline{S}_i$ and $\underline{M}_i$ are defined to represent the state of the squaring and multiplicative devices respectively after i clock pulses. Initially, the binary representation of α is fed into the squaring device and a 1 is put in the multiplicative circuit. The recursive nature of the algorithm becomes apparent if $\underline{S}_i$ is interpreted as follows:

$$\underline{S}_i = (\underline{S}_{i-1})^2 = \alpha^{2^i} \qquad (8)$$

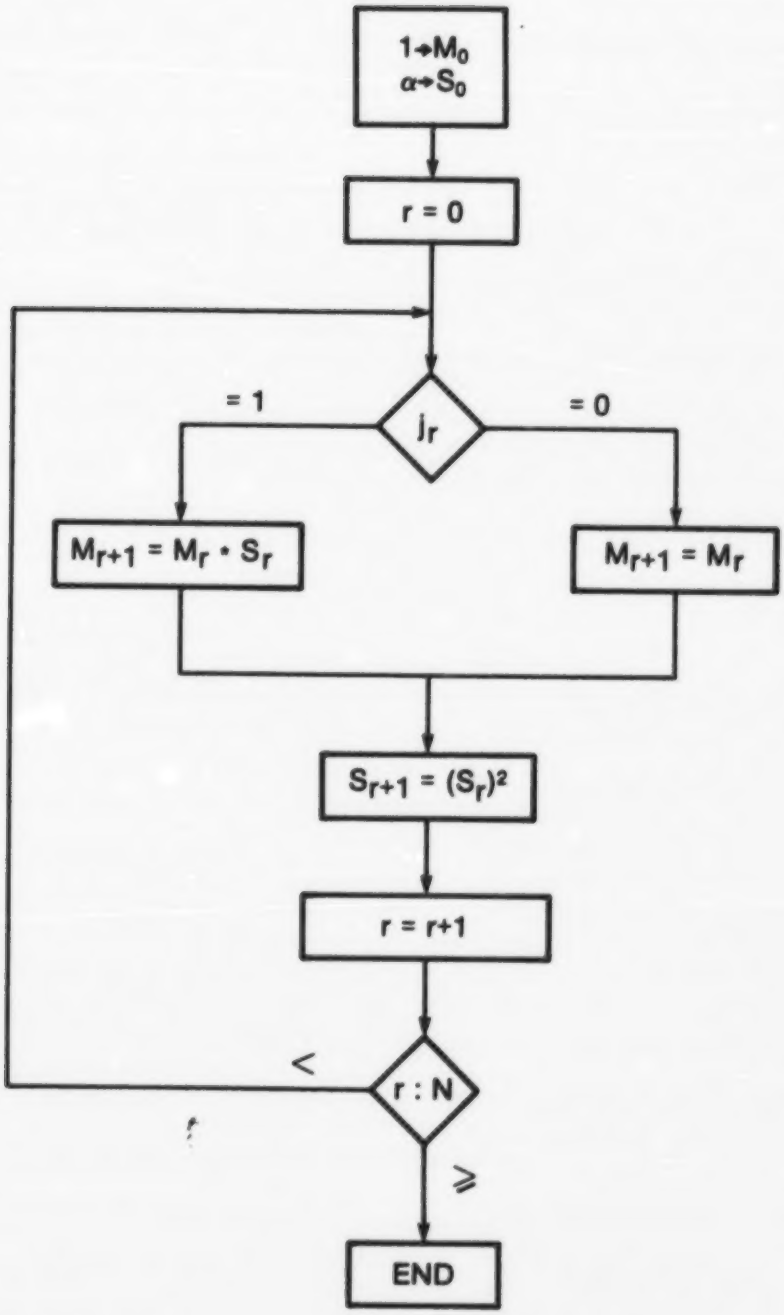


Figure 2 Simple algorithm to evaluate the binary coordinate of α^J. This may easily be implemented on a hardware circuit as described in the appendix A.

The mathematical representation of the algorithm is then given by equation 9 which forms the basis of the method.

$$\underline{M}_i = \prod_{r=0}^{i-1} \left(\alpha^{2^r} \right)^{j_r} = \prod_{r=0}^{i-1} (\underline{S}_r)^{j_r} = \underline{M}_{i-1} * (\underline{S}_{i-1})^{j_{i-1}} \qquad (9)$$

The process is terminated after N-clock pulses. At this point, the vector $\underline{M}$ correspond to α^J. The non zero elements of the vector $\underline{M}$, or equally α^J, indicates which stages of the shift register should be modulo-2 added to produce the desired result, the sequence delayed by J clock cycles, x^J.

To illustrate the concepts presented, an example using the generating polynomial $P(x) = 1 + x + x^4$ has been partially developed in appendix A.

4.0 APPLICATIONS

The method proposed in this document may serve a useful purpose in a number of applications. This section will be devoted to the brief description of three of them having interesting features in a military environment. These are: spread spectrum synchronization, communication protection and system reset.

4.1 Spread spectrum synchronization

In spread spectrum communication, one of the first concerns that must be dealt with in the process of retrieving the information from the received signal is the synchronization of the reference PN code to the modulating code, the alignment of the two code sequences. The traditional approach used for the alignment process consists of increasing the frequency of the clock of the reference code sequence until perfect alignment with the modulating code is accomplished. It is easily seen that the time required to complete such a procedure depends on the magnitude of the initial misalignment. The method proposed here would eliminate this drawback since the time required to complete the transition no longer depends on the size of the initial misalignment. The method could be used as a first step of the synchronization process, to set the reference sequence in the vicinity of the modulating code state. Then a higher frequency clock can be used to search for perfect alignment in a much smaller sequence region. This could become an important factor for receivers who were turned off for a long period of time.

4.2 Communication protection

The ECCM advantage introduced by the use of a PN code (direct sequence modulation or frequency hopping) is effective as long as unwanted users are unable to synchronize on the code sequence. To render the acquisition problem more difficult to undesirable listeners, it is possible to divide the PN code into short subsequences and to modulate the information stream with non consecutive subsequences, to introduce a non linear operation. The occurrence of these changes and their value could be established using either a predetermined scheme or the output of a second pseudorandom code generator. In any event, these changes have to be executed

very rapidly in order to maintain synchronization and to keep the flow of information continous. The approach presented here would enable this process to be implemented.

4.3 System reset.

Occasionally, in a spread spectrum system, a reset of the code sequence will be needed. This procedure may result from different factors such as: atomic clock frequency upset or simply a need to reboot the code sequence in a new state. In any of these events, a mechanism enabling the PN code to reach the required sequence state is necessary. This procedure has to be performed rapidly (especially after a clock offset) to preserve the communication link. The method developed here could help to meet this requirement.

5.0 CONCLUSION

Presented in this paper is a new method to rapidly access any one of the 2^N-1 possible delayed versions of a m-sequence. The result may be obtained within N-clock pulses, where N is the degree of the generating polynomial and consequently the number of stages in the shift register. The method may be easily implemented on hardware logic at low cost, only a squaring and a multiplicative circuits are required. The method can be used in many applications and can certainly offer great advantages to spread spectrum systems.

APPENDIX A

This appendix will demonstrate through the use of an example, the implementation of a delayed replica of a m-sequence. The example is based on the characteristic polynomial $P(x) = 1 + x + x^4$.

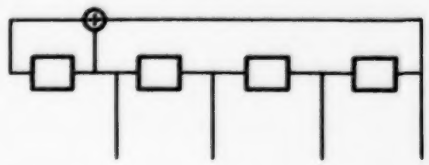
Table A1 shows the listing of two pseudorandom generators one can use to produce this maximum length sequence of period 15. These two generators are the Galois field generator, and the Fibonacci generator.

For this example tap #1 of the Fibonacci generator is chosen as the output reference for the m-sequence. If at time t=0 the shift register is loaded with α^0 , i.e. 1000, the sequence produced will be:

111101011001000...

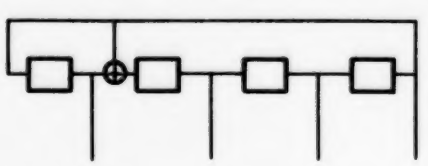
where the sequence goes from left to right.

Suppose now that after the seventh clock pulse it is desired to shift the output sequence back to its original position, at time t=0. After the seventh clock cycle, table A1 shows that the content of the Fibonacci generator reads 1010. Table A1 also shows from the binary value of α^7 of the Galois field GF(2^4) that taps 0, 1 and 3 of the Fibonacci generator should be combined in a modulo-2 adder to shift the sequence backward by 7 clock cycles. Figure A1 shows the MLS generator in this configuration. As shown in table A2, the output sequence presents at this time a 1, which is similar to the output produced at time t=0. Table A2 also shows the content of



Fibonacci generator

Taps #	0	1	2	3
States #				
0	1	0	0	0
1	1	1	0	0
2	1	1	1	0
3	1	1	1	1
4	0	1	1	1
5	1	0	1	1
6	0	1	0	1
7	1	0	1	0
8	1	1	0	1
9	0	1	1	0
10	0	0	1	1
11	1	0	0	1
12	0	1	0	0
13	0	0	1	0
14	0	0	0	1



Galois field generator

Taps #	0	1	2	3
States #				
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	1	0	0
5	0	1	1	0
6	0	0	1	1
7	1	1	0	1
8	1	0	1	0
9	0	1	0	1
10	1	1	1	0
11	0	1	1	1
12	1	1	1	1
13	1	0	1	1
14	1	0	0	1

Table A1 Partial listing of a Fibonacci generator and the corresponding Galois field generator for $P(x) = 1 + x + x^4$.

the generator and the output produced for the next 8 clock pulses. The pattern of the shift register content is not modified but the output sequence becomes similar to the output of an identical generator delayed by seven clock pulses. One can easily realize, however, the extensive work involved to produce table A1 for a complex polynomial or large delays.

With the use of the proposed method, the evaluation of α^7 required to produce a delay of 7 units in the sequence is easily computed. The binary

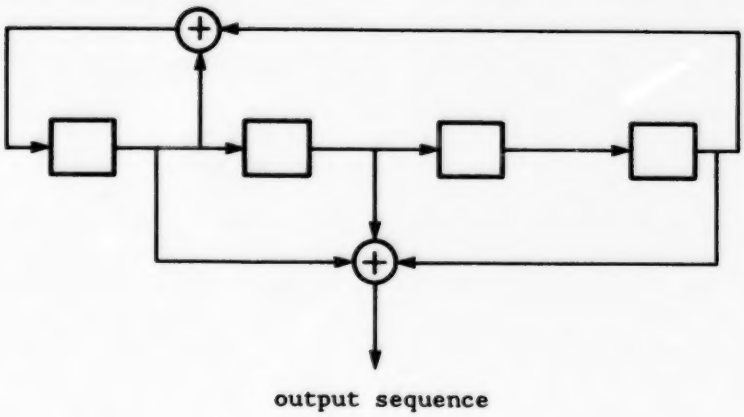


Figure A1 Configuration of a Fibonacci generator used to delay the sequence by 7 units.

State of the Fibonacci generator				output sequence
1	0	1	0	1
1	1	0	1	1
0	1	1	0	1
0	0	1	1	1
1	0	0	1	0
0	1	0	0	1
0	0	1	0	0
0	0	0	1	1
1	0	0	0	1
	etc			etc

Table A2 Partial listing of the effect generated by the configuration of figure A1. The sequence was previously clocked by seven pulses and then shifted back to the original position, at time t=0.

representation of 7 is 0111. This indicates that, starting with the initial condition α in the squaring device and 1 in the multiplicative circuit, α^2 will be produced in the squaring device and α in the multiplicative device after one clock pulse. After a second pulse, the content of the squaring device will read α^4 while that of the multiplicative, α^3 . Finally, after the third clock pulse, α^7 will be available in the multiplicative register. The fourth and last pulse required for this polynomial degree does not affect the operation since the fourth bit of the binary representation of 7 is zero.

The mathematical approach to the design of the squaring device will now be illustrated. Let ß be an element of $GF(2^4)$:

$$ß = a_0 + a_1 \alpha + a_2 \alpha^2 + a_3 \alpha^3 \quad , a_i \xi GF(2)$$

Squaring ß gives

$$ß^2 = a_0 + a_1 \alpha^2 + a_2 \alpha^4 + a_3 \alpha^6 \quad , \mathrm{mod}(1 + \alpha + \alpha^4)$$

which may be rewritten as:

$$ß^2 = (a_0+a_2) + a_2 \alpha + (a_1+a_3) \alpha^2 + a_3 \alpha^3$$

The initial state of the squaring device (a_0, a_1, a_2, a_3) is transformed into the final state $(a_0+a_2, a_2, a_1+a_3, a_3)$ after one clock pulse. By initializing the shift register of the squaring device with α , α^2 is being produced after one clock pulse and α^4 after two clock pulses and so on. Figure A2 shows the configuration of the squaring device for $P(x) = 1 + x + x^4$.

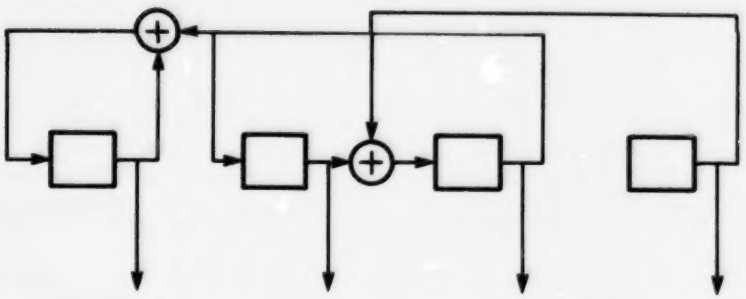


Figure A2 Representation of the squaring device for the generating polynomial $P(x)=1+x+x^4$. The tap connections give the value of the different power of α at each clock pulse.

A similar approach can be easily extended to the multiplicative circuit. Defining $A = a_0 + a_1 \alpha + a_2 \alpha^2 + a_3 \alpha^3$ to represent the previous result of the multiplication (initially 1) and $B = b_0 + b_1 \alpha + b_2 \alpha^2 + b_3 \alpha^3$ the content of the squaring device, $A*B \bmod(1+ \alpha + \alpha^4)$ is computed to show the required connections to implement the multiplicative circuit. The computation gives in this case:

$$c_0 + c_1 \alpha + c_2 \alpha^2 + c_3 \alpha^3$$

where

$$c_0 = a_0b_0 + a_1b_3 + a_2b_2 + a_3b_1$$

$$c_1 = a_0b_1 + a_1b_0 + a_1b_3 + a_2b_2 + a_2b_3 + a_3b_1 + a_3b_2$$

$$c_2 = a_0b_2 + a_1b_1 + a_2b_0 + a_2b_3 + a_3b_2 + a_3b_3$$

$$c_3 = a_0b_3 + a_1b_2 + a_2b_1 + a_3b_0 + a_3b_3$$

Cross terms are performed in hardware logic with AND gates and modulo-2 additions with exclusive-OR gates. According to the value of the binary element of the exponent J, j_k, the c terms are either fed back into the corresponding stage of the shift register defined by A or discarded.

With the aid of these basic elements, the logic implementation of the algorithm detailed in figure 2 is quite straightforward. The multiplication of the different power of α would only be executed when the binary values of the shift J would be different from zero.

Since the squaring device and the multiplicative device are characteristic of the polynomial used and not of the delay, they need to be evaluated only once. After the initial procedure is completed and the logic circuitry implemented, any delay can be performed given the binary representation of J. Figure A3 shows a generic design of the circuit needed to evaluate the vector α^J of a generating polynomial of degree 4.

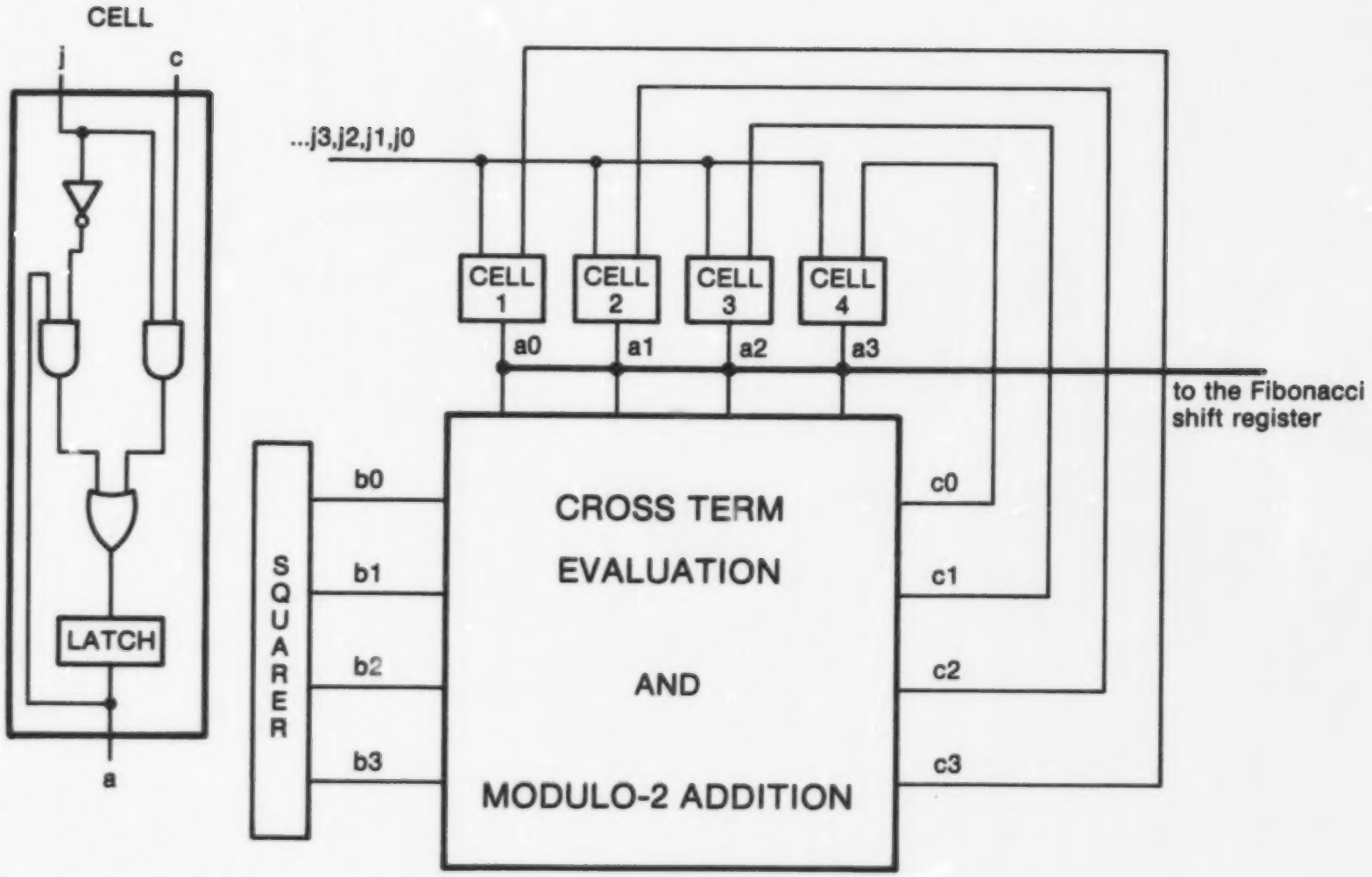


Figure A3 Generic design of the circuit required to produce the α^J vector for a polynomial of degree 4.

BIBLIOGRAPHY

[1] R.C. Dixon, "Spread Spectrum System," New-york, Wiley, 1976.

[2] J.J. Spilker,"Digital Communication by Satellite, "New-Jersey, Prentice-Hall, 1977.

[3] M.K. Simon, J.K. Omura, R.A. Scholtz, and B.K. Levitt, "Spread Spectrum Communications Vol.1," Computer Science Press, 1985.

[4] S.H. Tsao, "Generation of delayed replicas of maximal-length linear binary sequences," Proc. Inst. Elec. Eng. Vol 111, Nov 64, p.1803-1806.

[5] A.C. Davies, "Delayed versions of maximal-length linear binary sequences," Electron. Lett., Vol. 1, No. 3, May 65, p.61-62.

[6] A.B. Gardiner, "Logic p.r.b.s. delay calculator and delayed version generator with automatic delay-changing facility," Electron. Lett, Vol. 1, No.3, May 65, p. 123.

[7] A.J. Miller, A.W. Brown, and P.W. Mars, "A simple technique for the determination of delayed maximal-length linear binary sequences," IEEE Trans. Comp. Vol. C-26, No. 8, Aug 77, p.808-810.

[8] K.P. Yiu, "A simple method for the determination of feedback shift register connections for delayed maximal-length sequences," Proc. IEEE, Vol. 68, No. 4, Apr 80, p. 537-538.

[9] I.F. Blake and J.W. Mark, "Code division for spread spectrum multiple access", University of Waterloo, DSS project No. 808-01-03, March 81.

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM
(highest classification of Title, Abstract, Keywords)

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defence Research Establishment Ottawa
Ottawa, Ontario, Canada, K1A 0Z4

2. SECURITY CLASSIFICATION (overall security classification of the document. including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title.)

Generation of Delayed Replicas of a Maximal-Length Sequence Using Hardware Logic (U)

4. AUTHORS (Last name, first name, middle initial)

Bélisle, Capt J. Claude and Maltais, LCol Pierre

5. DATE OF PUBLICATION (month and year of publication of document)

Aug 87

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)

24

6b. NO. OF REFS (total cited in document)

9

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Note

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

PM E"F SATCOM, Electronics Division
Defence Research Establishment Ottawa
Ottawa, Ontario, Canada K1A 0Z4

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)

D6470-0459B

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DREO TECHNICAL NOTE 88-08

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(XX) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both offical languages unless the text is bilingual).

Pseudonoise (PN) sequences of maximal-length are widely used in a number of applications in communication systems. Very often a need arises to rapidly access any position within the sequence. Moreover, it is important to be able to predict when the move will be completed once it is initiated. Presented in this paper is a method that will give access to any one of the 2^n-1 positions of a sequence within n clock pulses, n being the number of stages in the shift register used to produce the sequence. The method is easily implemented on hardware logic circuitry and requires no software operation. The characteristic may help reduce the cost of operation.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

PN sequence
m - sequence
Spread Spectrum Synchronization
Spread Spectrum Systems
Galois Field

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM